

PI SETUP

Install image

Install the image by following the instructions from [raspberrypi.org](https://www.raspberrypi.org). Choose the RASPBIAN JESSIE LITE image, it's more than enough for our purposes.

You can now login if you connect a keyboard and monitor. The default user is `pi` with password `raspberry`.

Enable ssh

To be able to login without having to connect a monitor and keyboard you'll need SSH

```
sudo raspi-config:
```

- 5 interfacing options
- P2 SSH, select
- Yes
- Ok
- Finish

Networking

For this assignment we're going to setup a so-called ad-hoc WiFi network. From your workstation you'll connect to this network, so you'll be able to ssh into the pi in an easy manner.

setup an ad-hoc network over WiFi

For an example of how to setup an ad-hoc network over WiFi, you could read this [article about wireless communication](#).

```
sudo vi /etc/network/interfaces, and replace its contents with this:
```

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    wireless-channel 1
    wireless-essid NedapUniversity
```

```
wireless-mode ad-hoc
```

The first 3 lines are for wired networking. The last part is for the ad-hoc network.

To make the new wifi config active without a reboot do:

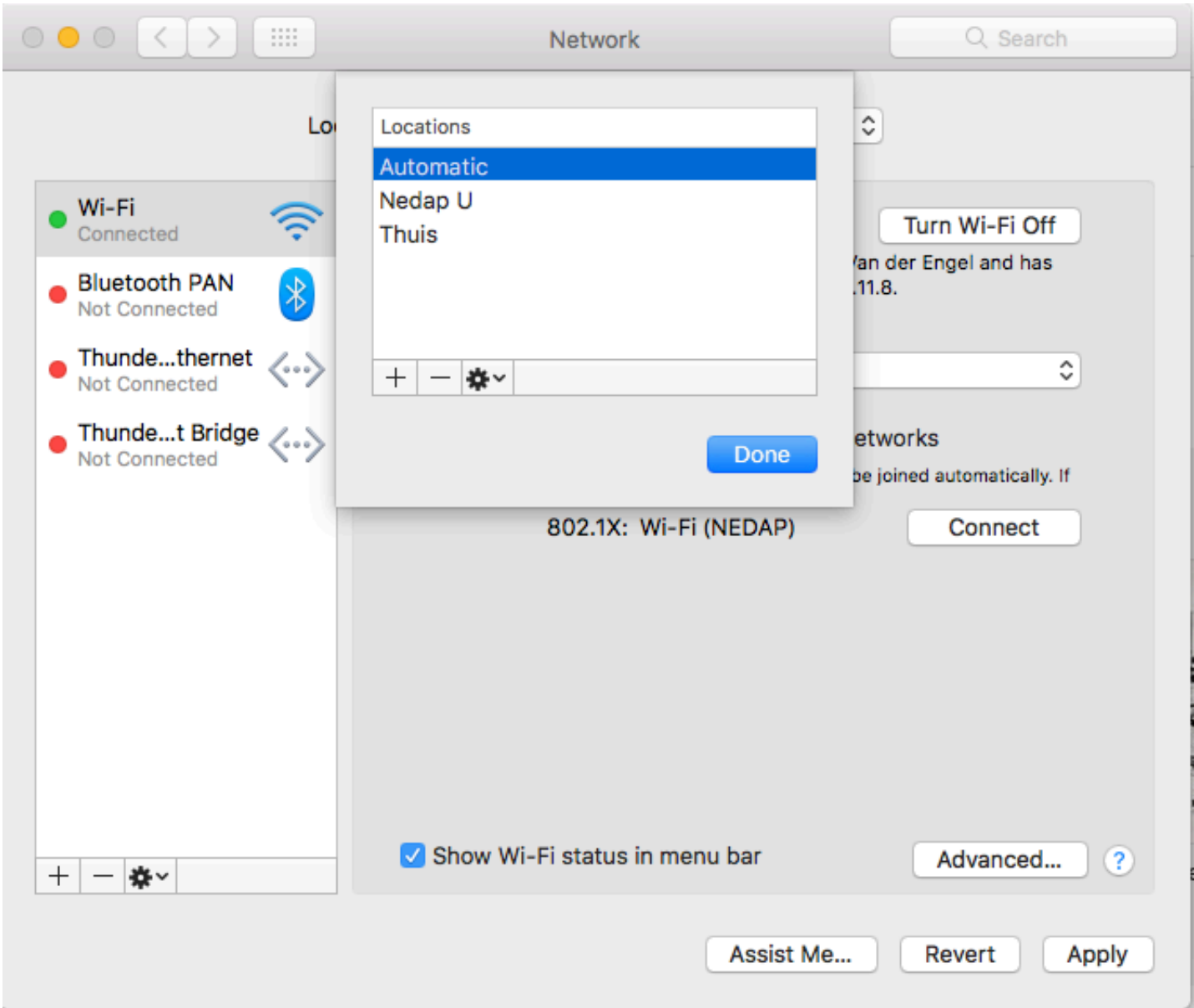
```
sudo ifdown wlan0  
sudo ifup wlan0
```

Connection

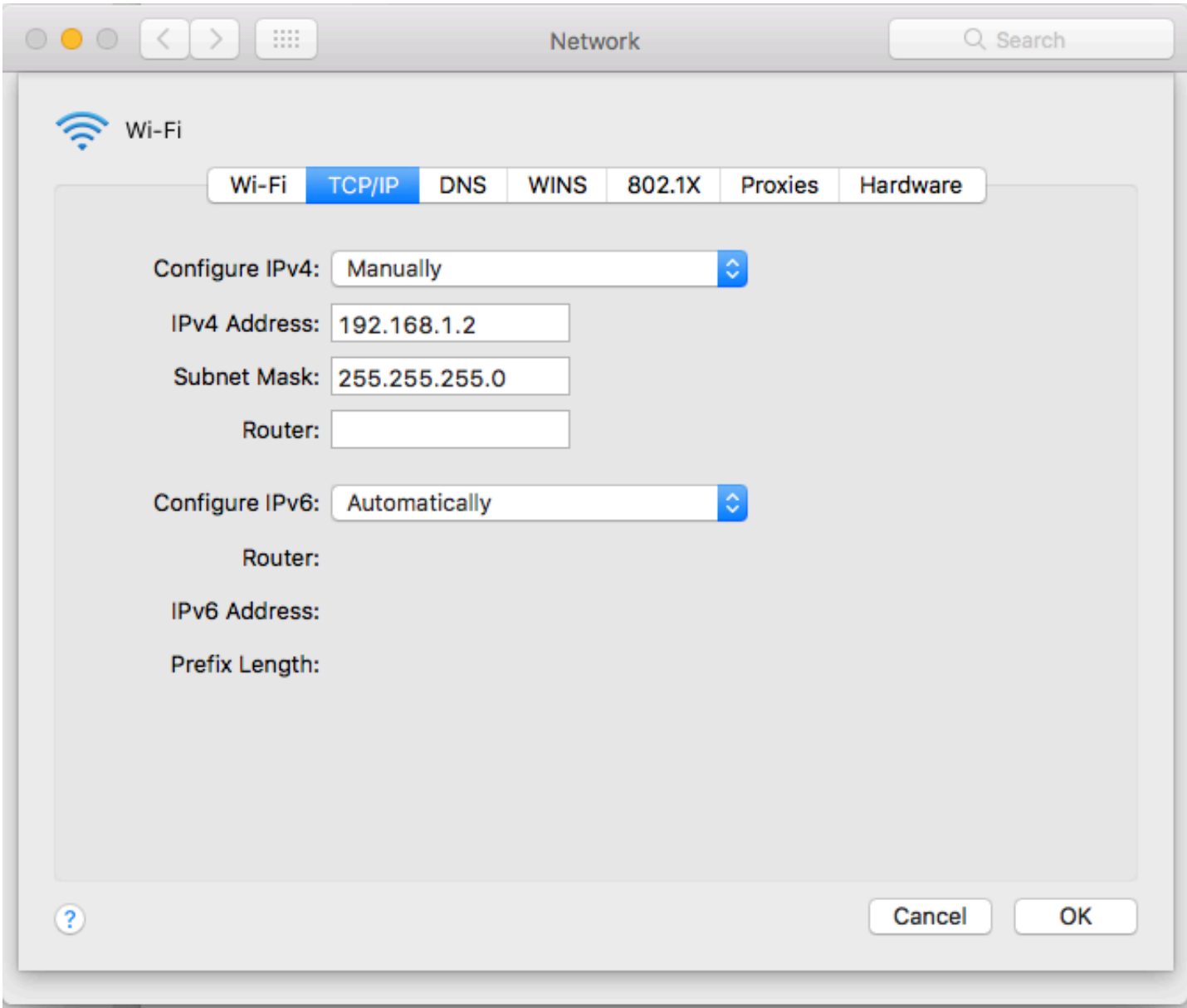
Connect your computer to this newly created ad-hoc network (Mac users; make sure to disable VPN security policy! and set your IP and subnet within range; 192.168.1.x and 255.255.255.0)

Mac setup

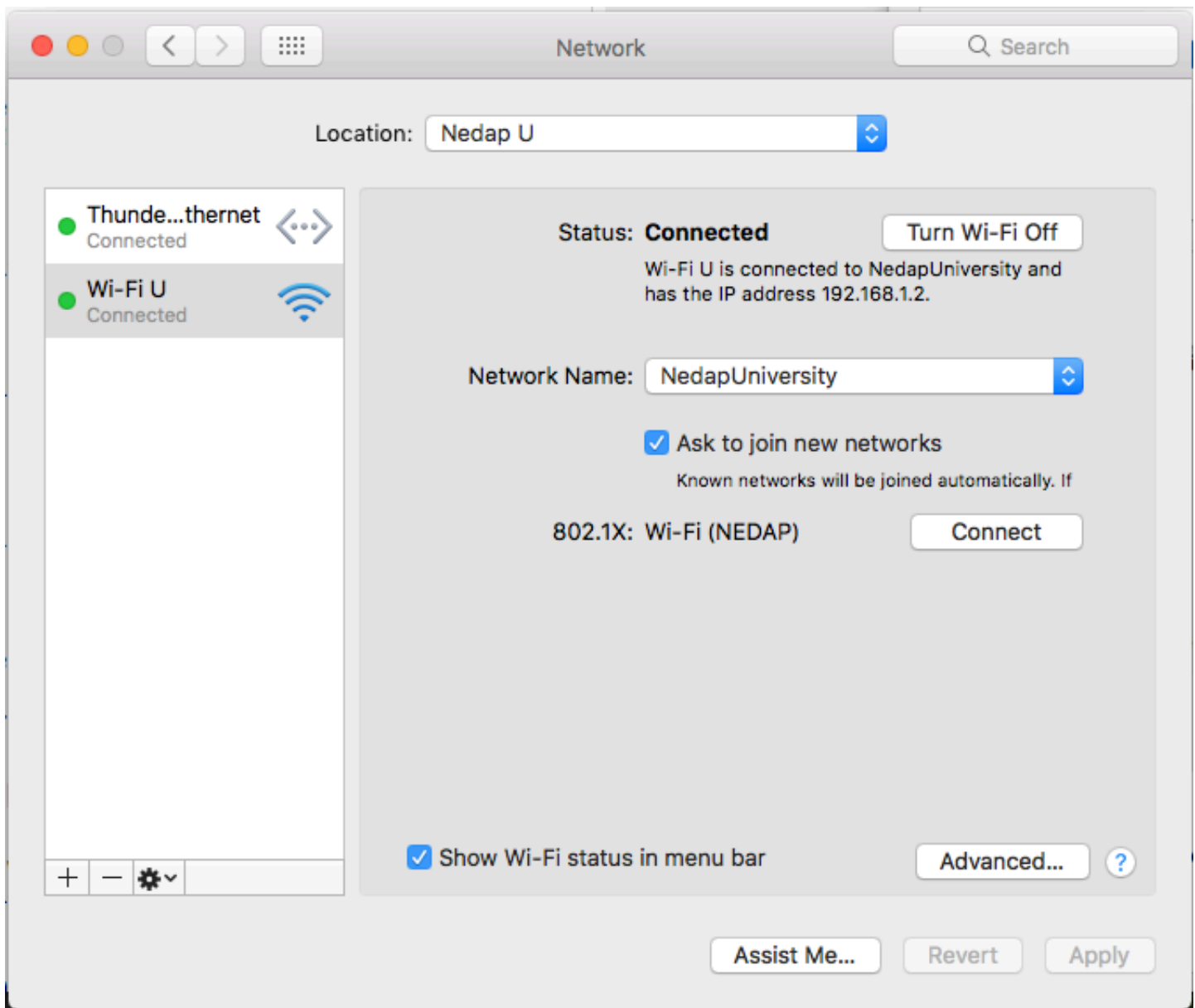
First add a new location, so you can easily switch configurations:



After that you should select the ad-hoc WiFi network and assign a manual IP address to your Mac.



This is what the final setup should look like:



You can now ssh from your Mac (Windows users can use PuTTY)

```
ssh pi@192.168.1.1 (password is "raspberr")
```

APPLICATION ENVIRONMENT

In order to be able to run Java programs, we'll need an JVM. This can simply be installed by using the standard installation tools: `sudo apt-get install oracle-java8-jdk`

Due to U.S. export regulations the JVM comes crippled, from a security standpoint. Therefore we have to replace some jar files within the JDK. First download the relevant zipfile from [Oracle's website](#). After that, unzip it and replace the `local_policy.jar` and `US_export_policy.jar` with the ones at `/usr/lib/jvm/jdk-8-oracle-arm32-vfp-`

hflt/jre/lib/security.

Project setup

The NUM2.zip contains a sample setup of Gradle project, that contains a task to automatically send a build jar to the Raspberry Pi over SSH.

Linux service

In order to start and stop our service when the Pi starts, we need a service wrapper. Create a new wrapper: `sudo vi /lib/systemd/system/num2.service` and paste this contents into it:

```
[Unit]
Description=Nedap U Service
After=multi-user.agent

[Service]
Type=simple
ExecStart=/usr/bin/java -jar /home/pi/NUM2.jar
Restart=on-abort
TimeoutStopSec=30

[Install]
WantedBy=multi-user.target
```

After doing a reload with `sudo systemctl daemon-reload`, the service can be started by invoking `sudo systemctl start num.service` and stopped with `sudo systemctl stop num.service`.

-
- <http://www.diegoacuna.me/how-to-run-a-script-as-a-service-in-raspberry-pi-raspbian-jessie/>